REMARKS

Claims 1-21 are pending and at issue. Claims 1-4, 6-8, 11-17 and 19-20 stand rejected as anticipated by Saeed. (USPN 6,711,447). The remaining claims each stand rejected under a purported combination of Saeed and Choi et al. (USPN 6,233,690). In light of the following remarks, Applicant respectfully asserts that the relied upon art does not teach or suggest the recited subject matter. Reconsideration of the rejection in light of the same is therefore respectfully requested.

The office action rejects claims 1-4, 6-8, 11-17 and 19-20 as anticipated by Saeed. In rejecting these claims, the office action takes the position that Saeed discloses an article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to: obtain from a performance monitor runtime performance data indicative of a *thread-level utilization*; and based on the performance data, adjust an operating voltage or an operating frequency of the machine.

However, Saeed actually discloses obtaining an indication of processor workload, defined by the level of *multi-threadedness* (col. 2, lines 7-13). Saeed discloses two methods for determining the workload. In the first, the operating system (OS) tracks the number of multi-thread threads that are running using a counter. In the second, counters track the percent of threads running that are spawned from multi-threaded applications. In this method, the workload is defined by the ratio of multi-threads to the total number of threads. (col. 2., lines 13-26). In either of these methods, the mere *number of threads* running on the system determines how CPU voltage and frequency are modulated. There is no modulation based on analysis of performance of any particular thread.

By contrast, claims 1-21 each recite an article or method in which runtime performance data indicative of *thread-level* utilization for a central processing unit (CPU) is obtained from a performance monitor. Thread-level utilization is distinguishable from both the number of threads and the ratio of multi-threads to total threads disclosed by Saeed. Thread-level utilization may be used to determine the performance characteristics or usage patterns of *individual threads running on the CPU.* Example data include, but are not limited to, instruction cache misses, data cache misses, instructions executed, branches executed,

branch mis-predicts, instruction translation look-up buffer TLB misses, data translation look-up buffer TLB misses, stalls due to data dependency, and data cache write-backs. By obtaining information about usage patterns of individual threads, the present invention provides for finer adjustment of operating voltage and frequency, adjustment during thread runtime, for example. Saeed would only measure overall thread presence, changing at most upon thread initiation or termination, not during execution.

With respect to various dependent claims, the office action rejects claim 3 as anticipated by Saeed, stating that Saeed discloses a performance monitoring unit that is part of a CPU. However, while Saeed does teach a power control circuit (PCC) coupled to one or more CPU cores (col. 3, lines 8-13), the PCC does not *monitor* runtime performance. Instead, the PCC receives a calculated workload from the OS and modulates voltage and/or frequency in response thereto. By contrast, claim 3 recites that the PMU itself is "part of a central processing unit (CPU) within the machine." Thus, it is clear that Saeed does not disclose a PMU that is part of a CPU, as recited by claim 3.

The office action also rejects claims 5, 9-10, 18 and 21 as obvious under Saeed in view of Choi et al. The prior art must teach or suggest all the claim limitations to establish a *prima facie* case of obviousness. *See, In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In this case, the prior art fails to teach all the claim limitations. Neither Saeed nor Choi et al., for example, teach obtaining from a performance monitor runtime performance data indicative of *thread-level utilization*, as recited by each of pending claims 1-21. The office action confirms that Saeed does not teach that additional runtime performance data includes cache misses and other data dependency stalls. As discussed above, Saeed teaches monitoring the number of multi-thread processes running at a given time, as opposed to monitoring of *thread-level* utilization, indicative of the performance of individual threads running at a given time. Likewise, Choi et al. do not teach obtaining data indicative of thread-level utilization for the purpose of voltage and frequency control of a functional unit. Instead of teaching monitoring data related to the performance of a thread, which may, for example, indicate processor usage patterns, Choi et al. teach monitoring data only to identify a stalled unit and then to block clock signals from being communicated to that unit until the stall is resolved.

The office action also takes the position, citing col. 1, lines 11-45, that Choi et al. teach that a computer can have its voltage or frequency adjusted due to the monitoring of performance data. However, Applicant is unable to find support for this position in the cited reference. Choi et al. appear to teach a technique for "clock gating," i.e., completely *decoupling* a clock signal from an electronic part when a trigger condition is detected. (col. 1, lines 24-27). Choi et al. does not appear to teach modulating the voltage or frequency of a functional unit, like the CPU. When the clock signal is decoupled from a device, "the affected part of the computer system is no longer charged and discharged." (col. 1, lines 27-29). This prevents the unit, the CPU in this case, from performing any calculations or executing any instructions. (*See*, col. 3, lines 7-10). Operation of the affected part is *suspended*, and processor execution logic is "powered down." (col. 2, lines 62-64). By contrast, *adjusting* the voltage and/or frequency of a CPU, would be understood as perhaps *changing the speed* at which a unit operates, but still allowing the CPU to continue to be available to process instructions or perform calculations, albeit at scaled speeds.

Further, Applicant is unable to find support for the position that Choi et al. disclose performance data consisting of an instructions-per-cycle metric, as recited by dependent claim 9. Yet, Choi et al. teach decoupling the clock from the processor (stopping execution, as described above) if *no* incoming instructions are detected. (col. 1, lines 40-45). Choi et al. do not teach modulating the processor voltage or frequency depending on how many instructions are being performed per cycle. Likewise, Choi et al. do not teach controlling voltage or frequency based on memory references-per-cycle metric, as recited in claim 10. For similar reasons, Choi et al. do not teach all the limitations recited by pending claim 21.

In view of the above remarks, Applicant respectfully requests reconsideration. All pending claims are in condition for immediate allowance.

Dated: August 17, 2006

Respectfully submitted,

By _____

Paul B. Stephens
Registration No.: 47,970
MARSHALL, GERSTEIN & BORUN LLP
233 S. Wacker Drive, Suite 6300
Sears Tower
Chicago, Illinois 60606-6357
(312) 474-6300
Attorney for Applicant